Medical Vision App - 5-Minute Video Script

INTRODUCTION (0:00 - 0:30)

[Screen: Show app logo and title]

Hello everyone! Today I'm excited to show you Medical Vision App, a powerful desktop application I developed for medical image analysis, specifically designed for CT scan processing.

This project integrates computer vision techniques, deep learning, and mathematical morphology to provide professional tools for anatomical segmentation and quantitative analysis of DICOM medical images.

Let's dive into what this application can do!

SECTION 1: LOADING AND VISUALIZATION (0:30 - 1:00)

[Screen: Demo loading a DICOM dataset]

The first feature is the ability to load and visualize complete DICOM datasets. The app uses ITK, the Insight Segmentation and Registration Toolkit, to natively read medical DICOM files.

[Action: Navigate through slices using the slider]

You can navigate through multiple slices using this intuitive slider, and the file list shows all available images in the dataset. The system automatically handles 16-bit medical images and converts them to 8-bit for visualization.

SECTION 2: IMAGE PREPROCESSING (1:00 - 1:45)

[Screen: Show preprocessing tab]

One of the most important features is the preprocessing pipeline. Medical images often contain noise, especially from low-dose CT scans, so we need to clean them up.

[Action: Enable DnCNN checkbox]

The app includes a DnCNN deep learning model for advanced denoising. This neural network was specifically trained for image denoising and can run either through a Flask server or locally using an ONNX model.

[Action: Show other filters]

Besides DnCNN, you can apply classical filters:

- Gaussian filter for smoothing
- Median filter for salt-and-pepper noise
- Bilateral filter that preserves edges

[Action: Enable CLAHE]

And for contrast enhancement, we have histogram equalization and CLAHE - Contrast Limited Adaptive Histogram Equalization, which is particularly effective for medical images.

[Action: Apply preprocessing and show results]

The system applies filters in the optimal order: first noise reduction, then contrast enhancement. After processing, you can see the PSNR and SNR metrics to evaluate image quality.

SECTION 3: ANATOMICAL SEGMENTATION (1:45 - 2:45)

[Screen: Switch to segmentation tab]

Now comes the core functionality: automatic anatomical segmentation. The app can detect three main structures in chest CT scans.

[Action: Click "Segment Lungs"]

First, lung segmentation. The algorithm uses adaptive thresholding based on Hounsfield Units, which are the standard measurement units in CT imaging. Lung tissue typically ranges from -1024 to -400 HU, corresponding to air-filled spaces.

[Screen: Show lung segmentation result in blue]

The system applies intelligent filtering based on anatomical position and area to eliminate small artifacts and ensure we're only detecting actual lung tissue.

[Action: Click "Segment Bones"]

Second, bone segmentation. This is more complex because it needs to detect multiple bone structures.

[Screen: Show bone segmentation result in white]

The algorithm includes a specialized sternum closing technique using mathematical morphology, and it filters out noise in the spinal column region. Bone detection uses the range of 400 to 3071 HU.

[Action: Click "Segment Aorta"]

Third, aorta segmentation, which detects the main blood vessel.

[Screen: Show aorta segmentation in red]

This works best with contrast-enhanced CT scans and uses the range of 120 to 300 HU. The algorithm performs connected component analysis and filters by area and anatomical position.

[Important note]

A key feature here is that you can choose to segment either the original image or the preprocessed version by checking this box. The system automatically adapts the thresholds based on the image type.

SECTION 4: MORPHOLOGICAL OPERATIONS (2:45 - 3:30)

[Screen: Switch to morphology tab]

After segmentation, you might want to refine the masks using morphological operations. The app implements seven fundamental operations:

[Action: Show dropdown menu]

- Erosion: reduces white regions
- Dilation: expands white regions
- Opening: removes small noise by eroding then dilating
- Closing: fills small holes by dilating then eroding
- Morphological Gradient: detects edges
- Top Hat: highlights small bright structures
- Black Hat: highlights small dark structures

[Action: Adjust kernel size and iterations]

You can configure the kernel size from 1 to 21 pixels and the number of iterations from 1 to 10.

[Action: Apply closing operation]

Let me demonstrate closing operation, which is useful for connecting nearby regions and filling small gaps.

[Action: Click "Fill Holes" button]

And there's also this dedicated button for filling holes using a flood fill algorithm, which is perfect for closing internal cavities in segmented organs.

SECTION 5: VISUALIZATION AND METRICS (3:30 - 4:15)

[Screen: Switch to visualization tab]

The visualization module lets you see all your segmented regions overlaid on the original image.

[Action: Toggle region checkboxes]

You can enable or disable each region independently and adjust the overlay opacity to see the underlying anatomy clearly.

[Screen: Switch to metrics tab]

Now, the metrics dashboard. This is where you get quantitative analysis.

[Action: Click "Update Metrics"]

The system automatically calculates:

- Area in pixels for each segmented structure
- Mean density in Hounsfield Units
- Signal-to-noise ratio per region
- PSNR compared to the original if you applied preprocessing

[Important feature: Ground Truth validation]

And here's a powerful feature for validation: you can load a Ground Truth mask, which is a manually segmented reference image.

[Action: Click "Load Ground Truth"]

The system calculates the Intersection over Union, or IoU metric, which measures the accuracy of the automatic segmentation.

[Screen: Show three-panel comparison window]

It opens this comparison window with three panels: automatic segmentation in blue, ground truth in green, and differences highlighted - white for matches, red for false positives, and yellow for false negatives.

SECTION 6: EXPORT FUNCTIONALITY (4:15 - 4:45)

[Screen: Show export button]

Finally, you can export all your results with a single click.

[Action: Click "Export Images"]

The export function saves all processing stages as PNG files:

- Original DICOM slice
- Preprocessed image
- Segmentation masks
- Colored overlays
- Morphological results
- Final visualization
- Even the histogram if available

[Screen: Show exported files in folder]

Everything is automatically named with the slice number and description, making it easy to organize your results for documentation or further analysis.

TECHNICAL OVERVIEW (4:45 - 5:00)

[Screen: Show code structure or architecture diagram]

From a technical perspective, the app is built with:

- C++ 17 for the core processing logic
- Qt 6 for the graphical interface
- OpenCV 4 for image processing and morphology
- ITK 6 for DICOM handling
- And Python with Flask for the DnCNN inference server

The entire project is open source and available on GitHub.

CONCLUSION (5:00 - 5:15)

[Screen: Show final results or application overview]

So that's Medical Vision App: a complete pipeline from DICOM loading, through preprocessing and segmentation, to morphological refinement and quantitative analysis.

It's designed to be a practical tool for medical image analysis, combining classical computer vision techniques with modern deep learning approaches.

Thank you for watching! If you want to try it yourself or contribute to the project, check out the GitHub repository linked in the description.

[Screen: Fade to GitHub URL and contact info]

Thanks again, and see you next time!

FILMING TIPS

Camera Setup

- Record in 1080p or higher
- Use good lighting (natural or ring light)
- Position camera at eye level
- Use external microphone for clear audio

Screen Recording

- Use OBS Studio or similar for screen capture
- Record at 1920x1080 resolution
- Zoom in on important UI elements
- Highlight cursor during actions
- Use screen annotations for emphasis

Editing Notes

- Add timestamps in video description
- Include zoom transitions for key features
- Add text overlays for technical terms
- Background music at low volume
- Color grade for consistency
- Add chapter markers:
  - 0:00 Introduction
  - 0:30 Loading DICOM
  - 1:00 Preprocessing
  - 1:45 Segmentation
  - 2:45 Morphology
  - 3:30 Visualization & Metrics
  - 4:15 Export

- 4:45 Technical Overview
- 5:00 Conclusion

B-Roll Suggestions

- Close-up of medical images
- Code snippets (briefly)
- Architecture diagrams
- Comparison before/after
- Export folder with results

Voice Over Tips

- Speak clearly and at moderate pace
- Pause between sections
- Emphasize key features
- Use enthusiastic but professional tone
- Practice beforehand for smooth delivery

Description Template for YouTube

```
Medical Vision App - Complete CT Image Analysis Tool

🔬 A desktop application for medical image processing and anatomical
segmentation of CT scans.

⏱ Timestamps:
0:00 Introduction
0:30 Loading DICOM Datasets
1:00 Image Preprocessing Pipeline
1:45 Automatic Anatomical Segmentation
2:45 Morphological Operations
3:30 Visualization and Metrics
4:15 Export Functionality
4:45 Technical Overview
5:00 Conclusion

🛠 Technologies Used:
- C++17 & Qt 6
- OpenCV 4.10
- ITK 6.0
- DnCNN Deep Learning
- Python Flask Server

🔗 Links:
GitHub: [Your Repository URL]
Documentation: [Link to README]

📧 Contact:
[Your Email]
```

```
#MedicalImaging #ComputerVision #ImageProcessing #DeepLearning #CTScans
#OpenCV #CPlusPlus
```